This article is divided into two sections

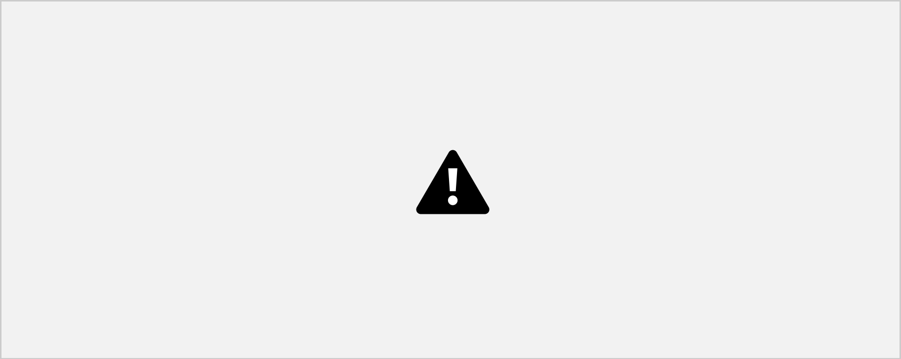
1. Interfacing PIR sensor to 8051
2. DIY intruder alarm circuit

## Interfacing PIR sensor to 8051 microcontroller

PIR sensors are widely used in motion detecting devices. This article is about interfacing a PIR sensor to 8051 microcontroller. A practical intruder alarm system using PIR sensor and 8051 microcontroller is also included at the end of this article. Before going into the core of the article, let’s have a look at the PIR sensor and its working.

### PIR sensor

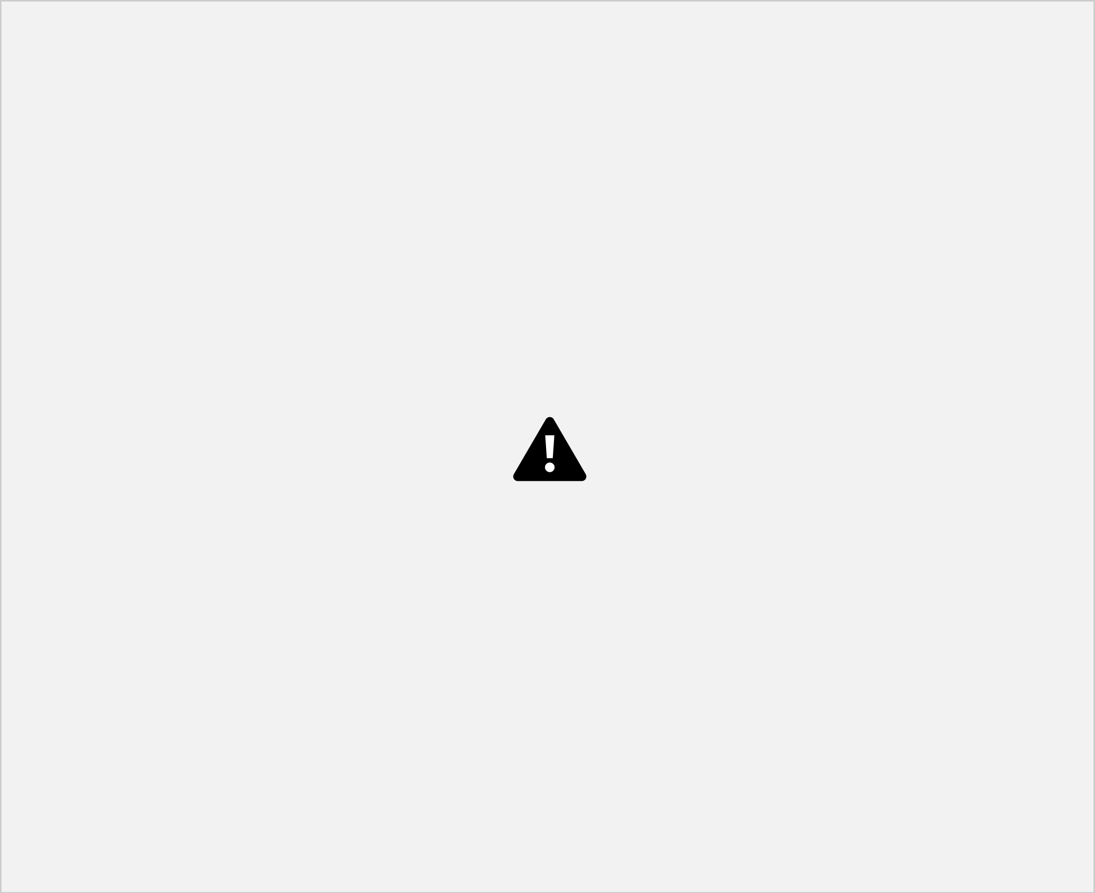
PIR sensor is the abbreviation of Passive Infrared Sensor. It measures the amount of infrared energy radiated by objects in front of it. They do not emit any kind of radiation but senses the infrared waves emitted or reflected by objects. The heart of a PIR sensor is a solid state sensor or an array of such sensors constructed from pyroelectric materials. The pyroelectric material is material by virtue of it generates energy when exposed to radiation. Gallium Nitride is the most common material used for constructing PIR sensors. Suitable lenses are mounted at the front of the sensor to focus the incoming radiation to the sensor face. Whenever an object or a human passes across the sensor the intensity of the of the incoming radiation with respect to the background changes. As a result the energy generated by the sensor also changes. Suitable signal conditioning circuits convert the energy change to a suitable voltage output. In simple words, the output of a PIR sensor module will be HIGH when there is motion in its field of view and the output will be LOW when there is no motion.



DSN-FIR800 is the PIR sensor module used in this project. Its image is shown above.  It operates from 4.5 to 5V supply and the standby current is less than 60uA. The output voltage will be 3.3V when the motion is detected and 0V when there is no motion. The sensing angle cone is 110° and the sensing range is 7 meters. The default delay time is 5 seconds. There are two preset resistors on the sensor module. One is used for adjusting the delay time and the other is used for adjusting the sensitivity. Refer the datasheet of DSN-FIR800 for knowing more.

## Interfacing PIR sensor to 8051

The 8051 considers any voltage between 2 and 5V at its port pin as HIGH and any voltage between 0 to 0.8V as LOW. Since the output of the PIR sensor module has only two stages (HIGH (3.3V) and LOW (0V)) , it can be directly interfaced to the 8051 microcontroller. The circuit diagram for interfacing PIR sensor to 8051 microcontroller is shown below.



The circuit shown above will read the status of the output of the PIR sensor and switch ON the LED when there is a motion detected and switch OFF the LED when there is no motion detected. The output pin of the PIR sensor is connected to Port 3.5 pin of the 8051. Resistor R1, capacitor C1 and push button switch S1 forms the reset circuit. Capacitors C3, C4 and crystal X1 are associated with the oscillator circuit. C2 is just a decoupling capacitor. LED is connected through Port 2.0 of the microcontroller. Transistor Q1 is used for switching the LED. R2 limits the base current of the transistor and R3 limits the current through the LED. Program for interfacing PIR sensor to 8051 is shown below.

### Interfacing Program

```
PIR EQU P3.5
LED EQU P2.0
ORG 00H
CLR P2.0
```

```
SETB P3.5
HERE:JNB PIR, HERE
     SETB LED
HERE1:JB PIR,HERE1
      CLR LED
SJMP HERE
END
```

The status of the output of the PIR sensor is checked using JNB and JB instructions. Code “HERE:JNB PIR, HERE” loops there until the output of the PIR sensor is HIGH. When it becomes HIGH it means a motion detected and the program sets P2.O HIGH in order to make the LED ON. The output pin of the PIR sensor remains HIGH for 5 seconds after a motion is detected. Code”HERE1:JB PIR,HERE1″ loops there until the output of the PIR sensor becomes LOW. When it becomes LOW the loop is exited and Port 2.0 is made LOW for switching OFF the LED. Then the program jumps back to label “HERE” and the entire cycle is repeated.

## DIY Intruder alarm using PIR sensor and 8051 microcontroller

This is just a serious practical application of PIR sensor and 8051 microcontroller. This circuit counts every intrusion and displays the number on intrusions on a 16×2 LCD display. An alarm is also made for 5 seconds on every intrusion.  A relay is used for switching the alarm buzzer. Additional loads like a bulb, solenoids etc can also be switched using the same relay. The circuit diagram of the intruder alarm using PIR sensor and 8051 microcontroller is shown below.

### Intruder alarm Circuit diagram

## Making

The circuit diagram of the PIR intruder alarm is shown above. The data pin D0 to D7 of the LCD module is connected to Port 0 of the microcontroller. The Port 0 of 8051 is an open drain and it will not work properly as an output port without an external pull up resistors. The resistor network R1 is used for pulling the Port 0 up. The control pins Rs, Rw and E of the LCD are connected to P2.7, P2.6 and P2.5 pins of the microcontroller. The output of the PIR sensor is connected to P3.5 of the microcontroller. P2.0 of the microcontroller is used for controlling the relay. Transistor Q1 is used for switching the relay. Resistor R7 limits the base current of the transistor. D5 is a freewheeling diode. R0 is a pull-up resistor. Since a relay is used for driving the buzzer you have the flexibility to use other loads like an electric bulb, solenoid, motor etc at the output instead of the buzzer. The program for interfacing PIR sensor to 8051 is shown below.

## Program

```
PIR EQU P3.5
RS EQU P2.7
```

```
RW EQU P2.6
E EQU P2.5
ORG 00H
MOV DPTR,#LUT
SETB P3.5
CLR P2.0
MOV R7,#00D
ACALL SPLIT
MAIN:ACALL DINT
ACALL TEXT1
ACALL LINE2
ACALL TEXT2
ACALL TEXT3
ACALL NUM


HERE:JNB PIR,HERE
SETB P2.0
INC R7
ACALL SPLIT
ACALL DINT
ACALL TEXT1
ACALL LINE2
ACALL TEXT4
ACALL TEXT3
ACALL NUM
HERE1:JB PIR,HERE1
CLR P2.0
SJMP MAIN

SPLIT:MOV A,R7
MOV B,#10D
DIV AB
MOV R6,B
MOV B,#10D
DIV AB
MOV R5,B
MOV B,#10D
DIV AB
MOV R4,B
RET
TEXT1: MOV A,#80D
ACALL DISPLAY
MOV A,#73D
ACALL DISPLAY
MOV A,#82D
ACALL DISPLAY
MOV A,#32D
ACALL DISPLAY
MOV A,#83D
ACALL DISPLAY
MOV A,#69D
ACALL DISPLAY
MOV A,#78D
ACALL DISPLAY
MOV A,#83D
ACALL DISPLAY
MOV A,#79D
ACALL DISPLAY
MOV A,#82D
ACALL DISPLAY
RET

TEXT2: MOV A,#83D
ACALL DISPLAY
MOV A,#67D
ACALL DISPLAY
MOV A,#65D
ACALL DISPLAY
MOV A,#78D
ACALL DISPLAY
MOV A,#73D
ACALL DISPLAY
MOV A,#78D
```

```
ACALL DISPLAY
MOV A,#71D
ACALL DISPLAY
MOV A,#46D
ACALL DISPLAY
MOV A,#46D
ACALL DISPLAY

RET

TEXT3: MOV A,#73D
ACALL DISPLAY
MOV A,#78D
ACALL DISPLAY
MOV A,#84D
ACALL DISPLAY
MOV A,#82D
ACALL DISPLAY

RET

TEXT4: MOV A,#65D
ACALL DISPLAY
MOV A,#76D
ACALL DISPLAY
MOV A,#69D
ACALL DISPLAY
MOV A,#82D
ACALL DISPLAY
MOV A,#84D
ACALL DISPLAY
MOV A,#33D
ACALL DISPLAY
MOV A,#33D
ACALL DISPLAY
MOV A,#33D
ACALL DISPLAY
MOV A,#33D
```

```
ACALL DISPLAY
RET

NUM:MOV A,R4
ACALL ASCII
ACALL DISPLAY
MOV A,R5
ACALL ASCII
ACALL DISPLAY
MOV A,R6
ACALL ASCII
ACALL DISPLAY
RET

DINT:MOV A,#0FH
ACALL CMD
MOV A,#01H
ACALL CMD
MOV A,#06H
ACALL CMD
MOV A,#83H
ACALL CMD
MOV A,#3CH
ACALL CMD
RET



LINE2:MOV A,#0C0H
ACALL CMD
RET

CMD: MOV P0,A
CLR RS
CLR RW
SETB E
CLR E
ACALL DELAY
```

```
RET

DISPLAY:MOV P0,A
SETB RS
CLR RW
SETB E
CLR E
ACALL DELAY
RET

DELAY: CLR E
CLR RS
SETB RW
MOV P0,#0FFh
SETB E
MOV A,P0
JB ACC.7,DELAY
CLR E
CLR RW
RET

ASCII: MOVC A,@A+DPTR
RET

LUT: DB 48D
DB 49D
DB 50D
DB 51D
DB 52D
DB 53D
DB 54D
DB 55D
DB 56D
DB 57D
```